

DO-NOTHING TEMPORARY DRIVER

Background of the Invention

Technical Field of the Invention

The present invention relates generally to the installation of drivers or the like in a computer system.

Background Art

FIG. 1 shows a system 10 which includes internet service provider equipment 12. The equipment 12 is of either a first type (type 1) or a second type (type 2). The internet service provider equipment is connected to a communication network 14 such as a telephone system or a cable system or other suitable communication means. There may be plural internet service providers and/or plural communication networks, but in the interest of simplicity and clarity, only a single example of each is shown.

Also connected to the communication network are one or more instances of user equipment 16. One 18 of the user equipments is shown in greater detail. The user equipment includes an operating system (OS) which has two or more possible interfaces (OS Interface 1, OS Interface 2) through which a service of the operating system may be accessed. In other words, a specific function of the operating system may be accessed via two or more different paths or ports or mechanisms. In the example shown, the operating system includes a network services mechanism (not shown) which other software can access via two or more interfaces (OS Interface 1, OS Interface 2), and the user equipment has been enhanced by the installation of a DSL card over which the user intends to access his internet service provider 12. Depending upon which type of back-end equipment the internet service provider uses, the user's DSL card will need to utilize either a first DSL driver (DSL Driver 1) or a second DSL driver (DSL Driver 2) to access the OS through the first or second interface, respectively. Typically, only one of the drivers and one of the OS interfaces will be installed at any given time.

FIG. 2 illustrates a typical method practiced in the prior art for installation and configuration of a system such as shown in FIG. 1. The user installs (20) the hardware, such as the DSL card. The OS, if it supports Plug and Play, will detect (22) the newly-installed DSL card at the next bootup, or perhaps immediately in the case of a DSL card and system which is hot plug ready. The OS will

launch (24) a Wizard which prompts the user to identify the location of the DSL driver to be used with that DSL card, whereupon the user may browse the computer's file system to locate the correct driver. Typically, the driver will be on a CD-ROM which will have separate drivers in separate directories for each of the possible OS interfaces that the DSL card may use, such as the PPPoA /
5 NDISWAN interface and the RFC 1483 Bridged Ethernet / NDIS interface.

Once the user has selected a driver, the Wizard installs (28) that driver. Then the user must manually configure (30) various settings in the OS, DLLs, and so forth. These may be scattered in various locations throughout the computer's systems, and may be accessed and set via a variety of different mechanisms. Typically, the user's manual will attempt to guide the user through this
10 process, albeit with historically poor efficacy.

Generally, the user will need to reboot (32) the system in order for the changes to take effect. If (34) the user was successful – and perhaps lucky, as well – the OS and its components will be complete and properly configured, the OS will now have the necessary components, upgrades, patches, and so forth to handle the selected driver, and the system will work correctly (36), including the newly-installed DSL card.

Frequently, however, the system will not be completely correctly configured, and (38) some part of the system will not operate, or will operate poorly. Generally, the computer will not provide the user with any indication of what is wrong, nor how to fix it. As one example, a frequent error is that the original OS installation did not include the DLLs or other components necessary to provide the functionality of the desired OS interface, and the user did not cause it to be installed during the process of installing and configuring the DSL card.

Brief Description of the Drawings

The invention will be understood more fully from the detailed description given below and
25 from the accompanying drawings of embodiments of the invention which, however, should not be taken to limit the invention to the specific embodiments described, but are for explanation and understanding only.

FIG. 1 shows one exemplary system as in the prior art.

FIG. 2 shows an exemplary flowchart of one method practiced in the prior art.

30 FIG. 3 shows an exemplary flowchart of one embodiment of a method of the invention.

FIG. 4 shows an exemplary embodiment of a system incorporating the invention.

Detailed Description

The invention is being described with reference to one specific embodiment, in which, upon the installation of a hardware device into a computer, the computer's operating system requires the installation of a driver software for the hardware device, and in which the operating system has two or more interfaces between itself and hardware devices of the general type to which the newly-installed hardware device belongs. However, the skilled reader will readily appreciate that the invention may be practiced in a variety of other embodiments, applications, and settings. For example, it may be used in the installation of a new software application, rather than in the installation of a new hardware device. Or, it may be used in conjunction with an application, rather than an operating system, having plural interfaces. Or, it may be used in a system which is not a conventional personal computer or the like, such as in a network router, or in an embedded control system, or other suitable system. For purposes of illustration and ease of explanation, the invention will be described in specific terms of a DSL card being installed into a PC running the Windows 98 operating system.

FIG. 3 illustrates one exemplary method practiced according to this invention. The user installs (50) the DSL card, and the OS detects (52) the DSL card at the next bootup or upon PNP detection.

The OS launches (54) a Wizard which, in some embodiments, may be a custom Wizard somewhat different in operation than the standard Wizard provided by the OS. The Wizard prompts (56) the user for the location of the DSL card's driver. However, rather than the user hunting through multiple alternative drivers in myriad possible locations, the user selects (58) a single common driver which may advantageously be stored in a commonly-accepted and widely-known location, such as in the root directory of the CD-ROM for example. This common driver may, in some embodiments, be a "do nothing" driver which does not necessarily accomplish anything actually related to the DSL card; rather, the function of the "do nothing" driver may be simply to satisfy the OS's demand for the user to identify the location of "the driver".

It is not necessarily the case that the custom, do-nothing driver accomplishes nothing. In some embodiments, it may perform temporary tasks, or it may perform one-time setup or configuration tasks which are common to the plurality of ultimate driver installations. In other embodiments, it may perform tasks which are not strictly necessary for the ultimate installation.

Furthermore, it is not necessarily the case that the custom driver or the ultimately-installed driver be a "driver" in the strictest Windows sense of the word. In some embodiments, each may be an information file (.INF file), some setup application files, documentation and help files, compressed driver files, a single file or multiple files, a Registry configuration, a virtual device driver file (.VXD file or .SYS file), a combination of those, and so forth. In other OS environments, each may be something different.

In the case of the Custom Wizard, it may advantageously create (62) a "Required Information Page" which gathers into a single location a list of data which the user needs to gather, and some indication of where these data may be found. Such data may include, for example: IP address, subnet mask, default gateway, DNS servers, OS interface type (all obtained from the ISP or the ISP's documentation), VPI/CPI, and driver encapsulation method (all obtained from the telco or the DSL provider). The user then gathers (64) the indicated data from the correct sources.

One of the pieces of data will typically be which type of OS interface the internet service provider's equipment expects to work with. The user may find that info from the printed materials that came with the DSL card, or by calling the internet service provider. Once the info is known, the user tells (66) the Wizard, typically by selecting from predetermined options, such as by picking one of two or more radio buttons in a conventional Windows dialog box or wizard page. The Wizard uninstalls (68) the "do nothing" common driver, and installs (70) the correct driver for the selected OS interface. One advantageous method is for the plural different drivers to be stored in compressed and/or encrypted form on the CD-ROM, with the Wizard knowing where each is stored and how to decompress/decrypt it. Decompression and/or decryption may individually or jointly be termed "decoding". The user is relieved of the responsibility of knowing where to find the right driver.

The Custom Wizard configures (72) the OS, DLLs, Windows Registry, and other such settings, according to the "Required Info" gathered by the user. The user is relieved of the responsibility of knowing where and how to set the various configurations according to this data.

If (74) the required OS interface is not installed, or the OS does not have all the necessary components, upgrades, patches, etc. to handle the selected driver, the Custom Wizard installs (76) the missing pieces. For example, if the internet service provider expects that the DSL card will access the OS via the PPPoA interface, but the corresponding NDIS5 software is not installed to enable NDISWAN access to the NDIS service of Windows, the Custom Wizard will install the

NDISWAN component of NDIS. The user is relieved of the responsibility of determining whether the OS is completely installed/upgraded/equipped.

While FIG. 3 has been described in terms of a method, it should also be understood to represent a machine-accessible medium having recorded or encoded or otherwise represented thereon instructions, routines, operations, control codes, or the like, which, when executed by or otherwise utilized by the machine, cause the machine to perform the method as described above or other embodiments thereof which are within the scope of this disclosure.

FIG. 4 illustrates a system 90 according to the invention. The user equipment includes the conventional operating system that has OS interfaces or sockets (only one of which is shown) and which makes a demand for the user to identify the driver for the newly-installed DSL card. The user equipment, along with other units of user equipment (which may or may not utilize the invention), are coupled via a conventional communication network to different types of internet service provider equipment (only a single type 1 is shown, for purposes of illustration).

The "do nothing" common driver is installed in response to the OS's driver ID demand, the custom wizard generates the required info page, the user gathers and provides the required information, and the custom wizard selects and installs the correct driver (in FIG. 4, this is DSL Driver 1 to go with the ISP equipment type 1, for example) and deinstalls the do nothing driver. Although the do nothing driver and the real driver are both illustrated in FIG. 4, the reader will appreciate that it is not necessarily the case that they both exist in the system at the same point in time, and that this and other aspects of FIG. 4 are merely for the purpose of teaching the invention and not limiting it.

While the invention has been described with reference to specific embodiments, the skilled reader will appreciate that it may be practiced in a wide variety of other embodiments. The reader will further appreciate that, for example, the relevant aspects of the illustrated operating system may be present in other types of software or such, which are not necessarily operating systems, yet the principles of this invention apply equally well in such cases. The reader will further appreciate that what has been discussed herein as a "wizard" may be implemented in a wide variety of formats, some of which may not fall into some definitions of "wizard". Similarly, what has been illustrated as a "driver" is intended to cover a variety of other instantiations.

Reference in the specification to "an embodiment," "one embodiment," "some embodiments," or "other embodiments" means that a particular feature, structure, or characteristic

described in connection with the embodiments is included in at least some embodiments, but not necessarily all embodiments, of the invention. The various appearances "an embodiment," "one embodiment," or "some embodiments" are not necessarily all referring to the same embodiments.

5 If the specification states a component, feature, structure, or characteristic "may", "might", or "could" be included, that particular component, feature, structure, or characteristic is not required to be included. If the specification or claim refers to "a" or "an" element, that does not mean there is only one of the element. If the specification or claims refer to "an additional" element, that does not preclude there being more than one of the additional element.

10 Those skilled in the art having the benefit of this disclosure will appreciate that many other variations from the foregoing description and drawings may be made within the scope of the present invention. Indeed, the invention is not limited to the details described above. Rather, it is the following claims including any amendments thereto that define the scope of the invention.